Mesh adaptation by local remeshing and application to immersed boundary methods in fluid mechanics.

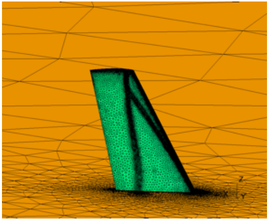
Workshop DIP

Léo Nouveau[1]
Héloïse Beaugendre[1,2], Mario Ricchiuto[1], Rémi Abgrall[3]
Cécile Dobrzynski[1,2], Algiane Froehly[1]
Pascal Frey[4], Charles Dapogny[5]

[1] CARDAMOM Team, INRIA Bordeaux Sud Ouest, France
[2] Bordeaux INP, IMB, France
[3] Institute of Mathematics & Computational Science, Zurich, Switzerland
[4] UPMC Paris 06, France
[5] Jean Kutzmann Lab, Grenoble, France

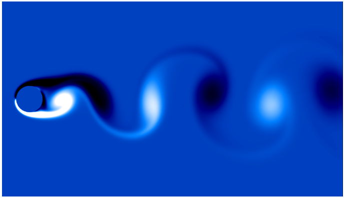
June 22, 2015

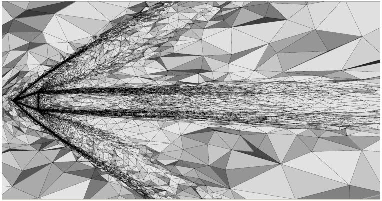
1. Introduction
2. MMG Platform
   - Overview
   - Metrics
   - MMGS
     - Presentation
     - Examples
   - MMG3D
     - Presentation
     - Exemples
3. Mesh adaptation for immersed boundary methods
   - Immersed boundary methods
   - Come back on mesh adaptation
   - Residual Distribution Schemes
   - Results
     - Steady Simulations
     - Unsteady simulations
4. Conclusion



- Need of adaptation in CFD :
  - Increase accuracy.
  - Limit time computation.

- Immersed Boundary Methods advantages for moving objects (ice shedding, windmill, ...) :
  - Not an explicit discretization on mesh.
  - Simplify mesh generation.
  - No constraint due to moving mesh.





- Open source platform : `https://github.com/MmgTools/mmg`
- Developpers : Cécile Dobrzynski, Algiane Froehly, Pascal Frey, Charles Dapogny.
- 3 Softwares :
  - MMGS : Surface remesher.
  - MMG2D : 2D remesher.
  - MMG3D : 3D remesher.
- Main assets :
  - Same data structure for the 3 softwares.
  - 4 developpers in 3 different institutes.

- Impose sizes and directions : metrics can be used[1].
- Positive Definite Symetric matrices.
- Can be defined for each node of the mesh.
- Contains size and direction of elements :

$$\mathcal{M} = {}^t\mathcal{R}\Lambda\mathcal{R},\ \ \mathcal{R} = (\mathbf{v}_1, \mathbf{v}_2),\ \ \Lambda = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}$$

  $\mathbf{v}_1, \mathbf{v}_2$: directions. Sizes are linked to the eigenvalues by : $h_i = 1/\sqrt{\lambda_i}$.
- Can be represented by an ellipse (in 2D, ellipsoid in 3D).

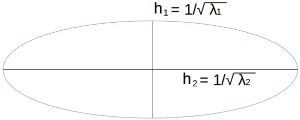
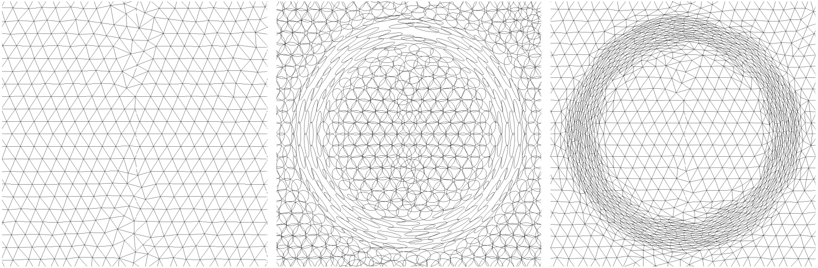


Figure : Representation of a metric in 2D

[1] P. Frey and F. Alauzet. "Anisotropic mesh adaptation for CFD computations". In: *Comput. Methods Appl. Mech. Engrg.* (2005).



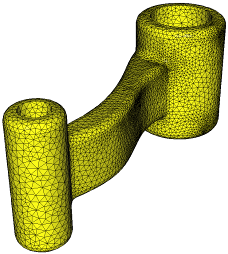
(a) Initial mesh (b) Metric imposition (c) Adapted mesh

Figure : Exemple of adaptation using metrics

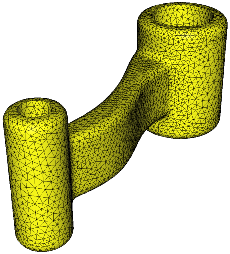
- Surface Remesher.
- Metric field given by the user.

- Surface Remesher.
- Metric field given by the user.
- Surface approximation improvement :
  - Ideal surface built on the initial mesh using 3^{rd} order Bezier triangles.
  - Control of the Hausdorff distance between ideal and mesh surface.

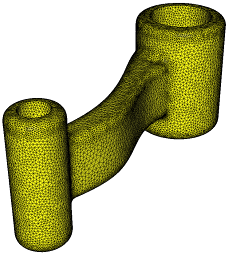
MMG Platform
MMGS - Examples



(a) initial mesh
22360 vertices



(b) hausdorff = 0.01
5629 vertices



(c) hausdorff = 0.001
22849 vertives

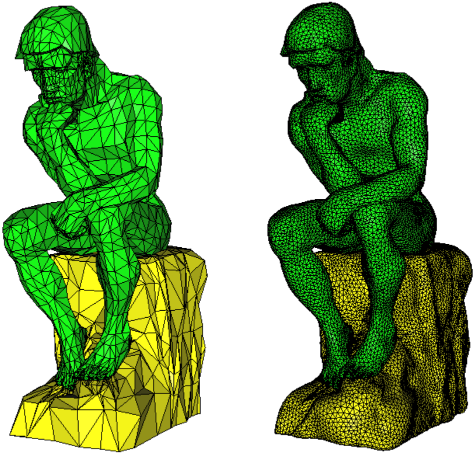
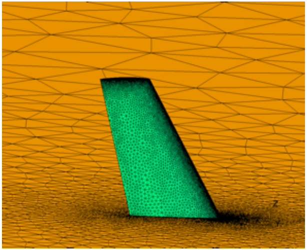
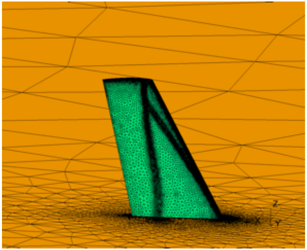


Figure : The Thinker of Rodin - hausdorff $= 3.10^{-3}$

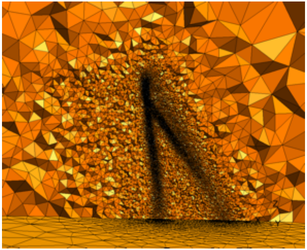
- 2 main goals :
  - 3D iso/aniso remeshing with domain remeshing.
  - Surface extraction based on a function on a 3D mesh.
- Software based on :
  - Metric tensors to prescribe sizes and directions of the edges.
  - Operator of local modification.
  - Geometric model based on 3^{rd} order Bézier triangles.
- 2 released versions :
  - MMG3D4.0 : Volumic adaptation iso/aniso.
  - MMG3D5.0 : Surfacic and iso volumic adaptation (current release : 5.0.1).



(a) Initial Mesh

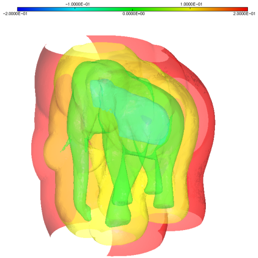


(b) Adaptation on the surface

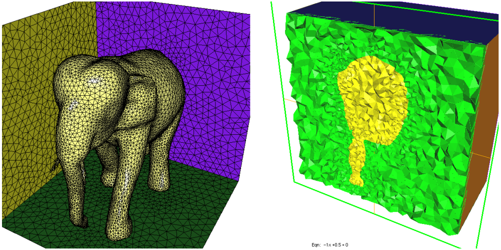


(c) Adaptation in the volume

Exemple surface extraction.



(a) Level set function defining the surface



(b) Mesh after extraction

Figure : Exemple of surface extraction

- Entire domain covered by a mesh.
- Not an explicit discretization of the solid.
- Use of level set function (signed distance function).
- Modification of the equations.

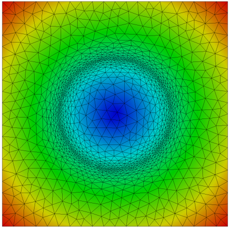
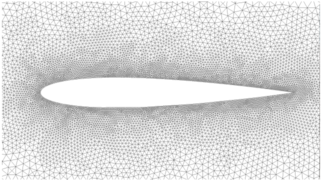
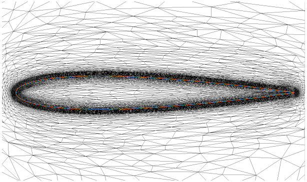


Figure : Signed distance function



(a) Fitted mesh



(b) Immersed boundary mesh

Figure : Mesh for different computations

Penalization : Penalty term accounts for Boundary conditions.

$$\begin{cases} \dfrac{\partial \rho}{\partial t} + \nabla.(\rho \mathbf{u}) = 0 \\ \dfrac{\partial (\rho \mathbf{u})}{\partial t} + \nabla.(\rho \mathbf{u} \otimes \mathbf{u}) + \dfrac{1}{\eta} \displaystyle\sum_{i=1}^{N_s} \chi_{s^i} (\rho \mathbf{u} - \rho \mathbf{u}_{s^i}) = -\nabla p + \nabla \underline{\boldsymbol{\pi}} \\ \dfrac{\partial (\rho e)}{\partial t} + \nabla((\rho e + p)\mathbf{u}) + \dfrac{1}{\eta} \displaystyle\sum_{i=1}^{N_s} \theta_{S^i} \chi_{s^i} (\rho \epsilon - \rho \epsilon_{s^i}) \\ \qquad + \dfrac{1}{\eta} \displaystyle\sum_{i=1}^{N_s} \chi_{S^i} (\rho \mathbf{u} - \rho \mathbf{u}_{S^i}).\mathbf{u} = \nabla(\underline{\boldsymbol{\pi}} \mathbf{u} + \mathbf{q}) \end{cases}$$

ρ density, $\mathbf{u}$ velocity, p pressure, $\underline{\pi}$ stress tensor, e total energy, ϵ internal energy, q heat flux, χ_{S^i} characteristic, θ_{S^i} to penalize or not the energy, $\eta << 1$ penalty parameter.

Penalization : Penalty term accounts for Boundary conditions.

$$
\begin{cases}
\dfrac{\partial \rho}{\partial t} + \nabla.(\rho \mathbf{u}) = 0 \\
\dfrac{\partial (\rho \mathbf{u})}{\partial t} + \nabla.(\rho \mathbf{u} \otimes \mathbf{u}) + \dfrac{1}{\eta} \displaystyle\sum_{i=1}^{N_s} \chi_{s^i} (\rho \mathbf{u} - \rho \mathbf{u}_{s^i}) = -\nabla p + \nabla \underline{\boldsymbol{\pi}} \\
\dfrac{\partial (\rho e)}{\partial t} + \nabla((\rho e + p)\mathbf{u}) + \dfrac{1}{\eta} \displaystyle\sum_{i=1}^{N_s} \theta_{S^i} \chi_{s^i} (\rho \epsilon - \rho \epsilon_{s^i}) \\
\qquad + \dfrac{1}{\eta} \displaystyle\sum_{i=1}^{N_s} \chi_{S^i} (\rho \mathbf{u} - \rho \mathbf{u}_{S^i}).\mathbf{u} = \nabla(\underline{\boldsymbol{\pi}} \mathbf{u} + \mathbf{q})
\end{cases}
$$

ρ density, $\mathbf{u}$ velocity, p pressure, $\underline{\pi}$ stress tensor, e total energy, ϵ internal energy, q heat flux, χ_{S^i} characteristic, θ_{S^i} to penalize or not the energy, $\eta << 1$ penalty parameter.

$$
\begin{cases}
\text{inside} : & \mathbf{u} = \mathbf{u}_s,\ \epsilon = \epsilon_s \\
\text{outside} : & \textit{Usual Navier-Stokes}
\end{cases}
$$

Aim

Reduction of the error by adaptation

Aim

Reduction of the error by adaptation

0 level set
Majoration of the error on the geometry

$$\mathcal{M} = {}^t\mathcal{R} \begin{pmatrix} \frac{1}{\epsilon^2} & 0 & 0 \\ 0 & \frac{|\lambda_1|}{\epsilon} & 0 \\ 0 & 0 & \frac{|\lambda_2|}{\epsilon} \end{pmatrix} \mathcal{R}$$

ϵ error, λ_i e. va. of the hessian of ϕ
$\mathcal{R} = (\nabla\phi \;\; v_1 \;\; v_2)$, (v_1, v_2) tangential plane of the surface.

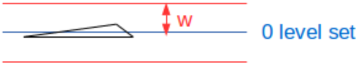


Figure : Insertion area of fine cells.

Aim

Reduction of the error by adaptation

0 level set
Majoration of the error on the geometry

$$\mathcal{M} = {}^t\mathcal{R} \begin{pmatrix} \frac{1}{\epsilon^2} & 0 & 0 \\ 0 & \frac{|\lambda_1|}{\epsilon} & 0 \\ 0 & 0 & \frac{|\lambda_2|}{\epsilon} \end{pmatrix} \mathcal{R}$$

ϵ error, λ_i e. va. of the hessian of ϕ
$\mathcal{R} = (\nabla\phi \;\; v_1 \;\; v_2)$, (v_1, v_2) tangential plane of the surface.

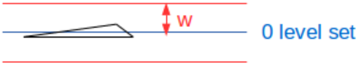


Figure : Insertion area of fine cells.

Physics
Majoration of the interpolation error

$$\mathcal{M} = {}^t\mathcal{R} \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix} \mathcal{R}$$

$\mathcal{R}$ e. ve. of the hessian of the solution,

$$\lambda_i = min\left(max\left(|h_i|, \frac{1}{h_{max}^2} \right), \frac{1}{h_{min}^2} \right)$$

h_{min} (h_{max}) min. and max. size,
h_i the e. va. of the hessian of the solution.

Steady Conservation Law

$$\nabla.\mathcal{F}(u) = 0$$

Steady Conservation Law

$$\nabla.\mathcal{F}(u) = 0$$

Fluctuation

$$\phi^T = \int_T \nabla.\mathcal{F}(u)$$

Steady Conservation Law

$$\nabla.\mathcal{F}(u) = 0$$

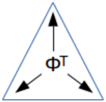
Fluctuation

$$\phi^T = \int_T \nabla.\mathcal{F}(u)$$

Ditribution to the degree of freedom

Nodal Residual

$$\phi_i^T = \beta_i^T \phi^T$$



Steady Conservation Law

$$\nabla.\mathcal{F}(u) = 0$$

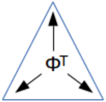
Fluctuation

$$\phi^T = \int_T \nabla.\mathcal{F}(u)$$

Ditribution to the degree of freedom

Nodal Residual

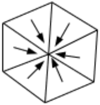
$$\phi_i^T = \beta_i^T \phi^T$$



Gathering the contribution of the triangle the point i belongs

Residual Distribution Scheme

$$\sum_{T|i\in T} \phi_i^T = 0$$



Resolution with pseudo time step

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + \frac{1}{|C_i|} \sum_{T|i \in T} \phi_i^T(u_h^{n+1}) = 0$$

Resolution with pseudo time step

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + \frac{1}{|C_i|} \sum_{T|i \in T} \phi_i^T(u_h^{n+1}) = 0$$

Algorithm for adaptation

1. Initial mesh adapted to 0 level set.
2. Computation.
3. Adaptation to : 0 level set + solution.
4. Interpolation of the solution on the new mesh.
5. Computation on new mesh from old solution.
6. Go to 3 or exit if convergence Mesh/Solution.

Unteady Conservation Law

$$\partial_t u + \nabla . \mathcal{F}(u) = 0$$

Unteady Conservation Law

$$\partial_t u + \nabla.\mathcal{F}(u) = 0$$

Total Residual

$$\Phi^T = \int_{t^n}^{t^{n+1}} \int_T (\partial_t u + \nabla.\mathcal{F}(u)) = \int_{t^n}^{t^{n+1}} \int_T (\partial_t u) + \Delta t \phi^T$$

Unteady Conservation Law

$$\partial_t u + \nabla.\mathcal{F}(u) = 0$$

Total Residual

$$\Phi^T = \int_{t^n}^{t^{n+1}} \int_T (\partial_t u + \nabla.\mathcal{F}(u)) = \int_{t^n}^{t^{n+1}} \int_T (\partial_t u) + \Delta t \phi^T$$

Distribution/Gathering

$$\sum_{T|i\in T} \underbrace{\left\{ \beta_i \int_T (u_h^{n+1} - u_h^n) + \Delta t \phi_i^T (u_h^*) \right\}}_{\Phi_i^T} = 0$$

- Choice for $* = n + \frac{1}{2}$:

$$u_h^* = u_h^{n+\frac{1}{2}} = \frac{u_h^{n+1} + u_h^n}{2} \tag{1}$$

- Unsteady resolution : explicit RK2-RDS[2]

$$\begin{cases} \textit{Step 1 : } \mathbf{U}_i^1 = \mathbf{U}_i^n - \dfrac{\Delta t}{|C_i|} \displaystyle\sum_{T|i\in T} \phi_i^T(\mathbf{U}_h^n) \\ \textit{Step 2 : } \mathbf{U}_i^2 = \mathbf{U}_i^1 - \dfrac{\Delta t}{|C_i|} \displaystyle\sum_{T|i\in T} \beta_i^T \int_T \left\{ \dfrac{\mathbf{U}_h^1 - \mathbf{U}_h^n}{\Delta t} + (\nabla.\mathbf{F} + \nabla.\mathbf{G}) \left(\dfrac{\mathbf{U}_h^1 + U_h^n}{2} \right) \right\} \end{cases}$$

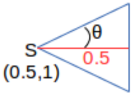
[2] M. Ricchiuto and R. Abgrall. "Explicit Runge-Kutta residual distribution schemes for time dependent problems: Second order case". In: *J. Comput. Phys. 229* (2010).

- Penalization in explicit : $\Delta t \sim \eta (= 10^{-10})$.
- Proposed scheme :

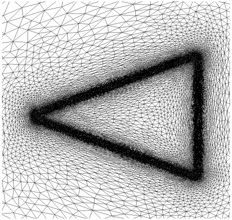
Explicit RK2-RDS for NS part + implicit splitting for penalization

$$\begin{cases} \textit{Step 1}: \mathbf{U}_i^1 = \mathbf{U}_i^n - \dfrac{\Delta t}{|C_i|} \displaystyle\sum_{T|i\in T} \phi_i^T(\mathbf{U}_h^n) \\ \textit{Step 2}: \mathbf{U}_i^2 = \mathbf{U}_i^1 - \dfrac{\Delta t}{|C_i|} \displaystyle\sum_{T|i\in T} \beta_i^T \int_T \left\{ \dfrac{\mathbf{U}_h^1 - \mathbf{U}_h^n}{\Delta t} + (\nabla.\mathbf{F} + \nabla.\mathbf{G}) \left(\dfrac{\mathbf{U}_h^1 + U_h^n}{2} \right) \right\} \\ \textit{Step 3}: \dfrac{\mathbf{U}_i^{n+1} - \mathbf{U}_i^2}{\Delta t} + \mathbf{S}(\mathbf{U}_i^{n+1}) = 0 \end{cases}$$

- Triangle placed in a circle of radius 20.
- Initial mesh adapted to the 0 level set.
- $Re = 50000$, $Ma = 2.366431913$, $p = 1/\gamma$.
- Initial mesh : 30407 vertices, adapted mesh : 111061 vertices



(a) Triangle



(b) zoom on initial mesh

Mesh adaptation for immersed boundary method
Steady Simulations - 2D Supersonic triangle

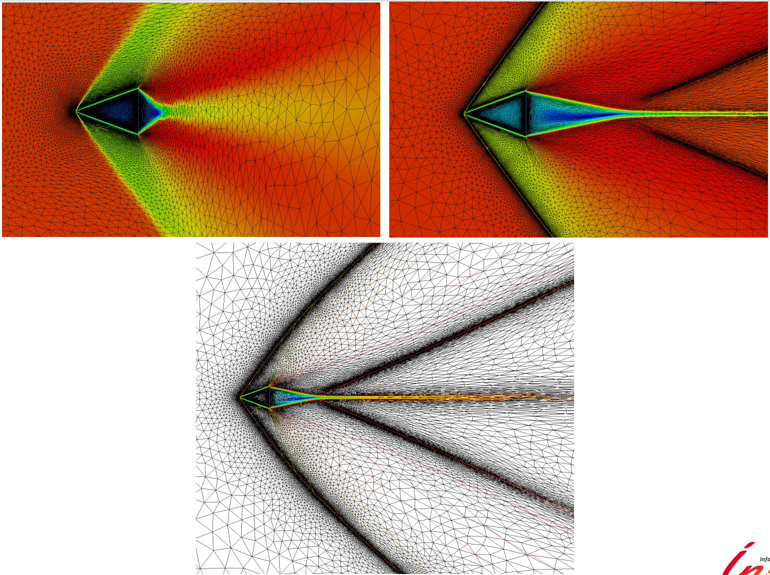
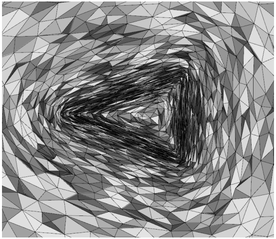


Figure : u velocity on initial and adapted mesh

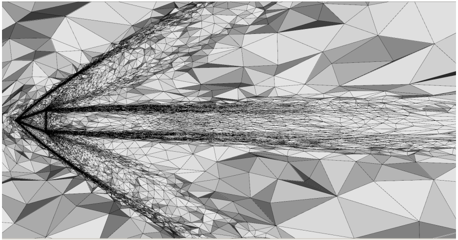
- Validation with literature[3] :
  - Angle between shock and $y = 0$ has analytical value : we find $\beta = 53.33\,\text{deg}$ for $\beta_{analytic} = 53.46\,\text{deg}$.
  - Cut of the pressure in agreement with the result of Boiron *et al.*
- Good approximation of the geometry thanks to mesh adaptation.
- Good capture of the physics (shock + drag) owing to mesh adaptation.
- Limitation of the number of nodes and elements.

[3] O. Boiron, G. Chiavassa, and R. Donat. "A High-Resolution Penalization Method for large Mach number Flows in the presence of obstacles". In: *J. Comput. F.* (2008).

- Same test case in 3D.
- Extrusion of the triangle in the z direction of 0.364.
- Sphere of radius 15.
- Initial mesh : 36597 vertices, adapted mesh : 766310 vertices.



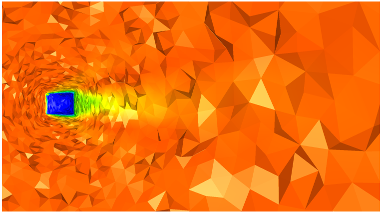
(a) Initial mesh



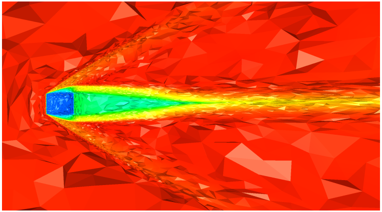
(b) Adapted mesh

Figure : $z = 0.182$ cut

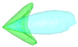
Mesh adaptation for immersed boundary method
Steady Simulations - 3D Supersonic triangle



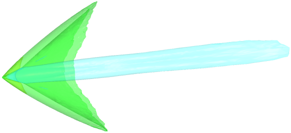
(a) u velocity on initial mesh
$y = 0$ cut



(b) u velocity on adapted mesh
$y = 0$ cut

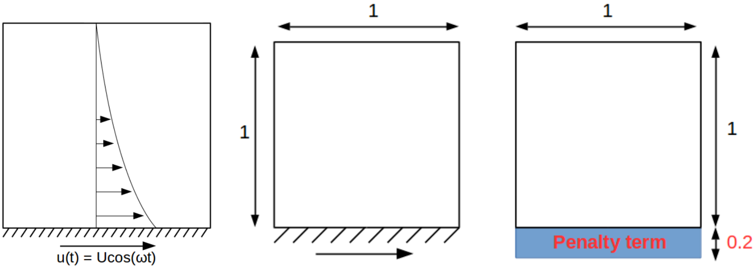


(c) density on initial mesh



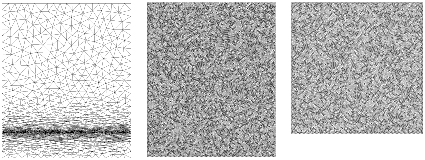
(d) density on adapted mesh

- Initially motionless fluid moved by wall.
- With hypothesis, analytical results for validation.
  - Constant speed : $u(y,t) = Uerf(\frac{y}{2\sqrt{\mu t}})$.
  - Oscillating wall : $u(y,t) = U\exp^{-ky}\cos(ky-\omega t), \quad k = \sqrt{\frac{\omega}{2\mu}}$



(a) Test case presentation

(b) Domain for non penalized and penalized simulations



(a) $1,115$ vertices (b) $16,053$ vertices (c) $13,449$ vertices

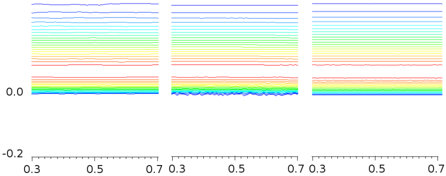
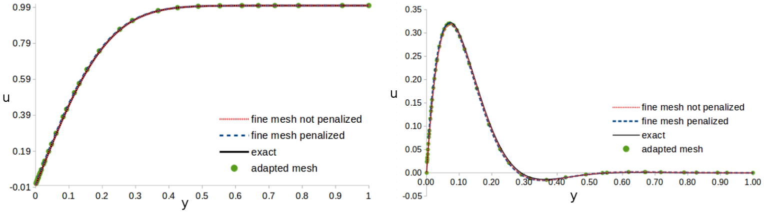


Figure : isoline of the u velocity on different mesh

Mesh adaptation for immersed boundary method

Unsteady Simulations - Rayleigh case



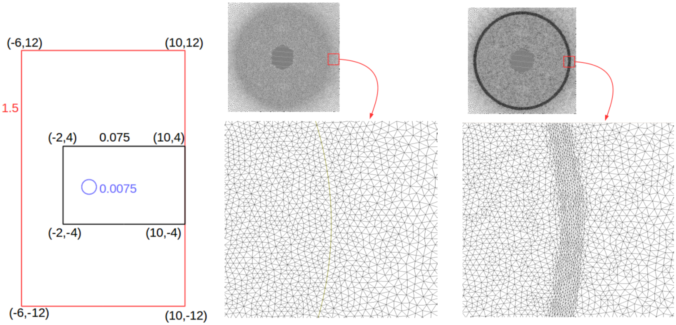
(a) Constant speed   (b) Oscillating wall

Figure : Comparison of solutions

- Mesh adaptation allow to have a better approximation of the wall with less elements than an very fine uniform mesh.
- Validation of the proposed method looking at the differences between the solutions pena/nopena and pena/analytic.

Mesh adaptation for immersed boundary method
Unsteady Simulations - Flow past cylinder

- Cylinder (radius $r = 0.5$) in a box $[-6, 10] \times [-12, 12]$.
- $Re = 200,\ Ma = 0.2,\ \rho = 1,\ p = 1/\gamma$.



(a) Domain

(b) Fitted mesh
67815 vertices

(c) Adapted mesh
85945 vertices

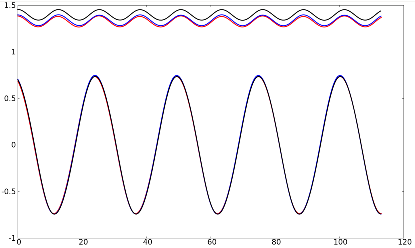
Two forces computations :

- Integration of pressure/shear stress over the edges discretizing the solid.
- Change of momentum : $\mathbf{F} = \frac{\Delta \mathbf{m}}{dt}$, $\quad \Delta \mathbf{m} = \int_S \rho(\mathbf{u} - \mathbf{u}_S)$.
- Comparison with literature[4].

[4] M. Bergmann, L. Cordier, and J.P. Brancher. "Optimal rotary control of the cylinder wake using proper orthogonal decomposition reduced-order model". In: *Phys. Fluids* 17.9, 097101 (2005).

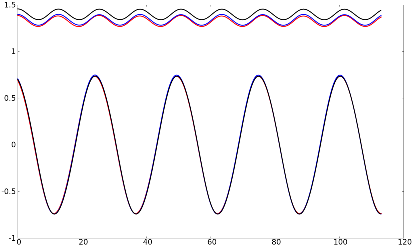
Two forces computations :

- Integration of pressure/shear stress over the edges discretizing the solid.
- Change of momentum : $\mathbf{F} = \frac{\Delta \mathbf{m}}{dt}, \qquad \Delta \mathbf{m} = \int_S \rho(\mathbf{u} - \mathbf{u}_S).$
- Comparison with literature[4].



Author	St	C_D
Braza *et al.*	0.2000	1.4000
Henderson	0.1971	1.3412
He *et al.*	0.1978	1.3560
Bergmann *et al.*	0.1999	1.3900
Fitted Mesh, "cc"	0.1965	1.3979
Fitted Mesh, "cmc"	0.1965	1.3404
Adapted Mesh	0.1965	1.3280

[4] M. Bergmann, L. Cordier, and J.P. Brancher. "Optimal rotary control of the cylinder wake using proper orthogonal decomposition reduced-order model". In: *Phys. Fluids* 17.9, 097101 (2005).



Author	St	C_D
Braza *et al.*	0.2000	1.4000
Henderson	0.1971	1.3412
He *et al.*	0.1978	1.3560
Bergmann *et al.*	0.1999	1.3900
Fitted Mesh, "cc"	0.1965	1.3979
Fitted Mesh, "cmc"	0.1965	1.3404
Adapted Mesh	0.1965	1.3280

- All computations : Lift and Strouhal number OK.
- Force computation by integration : Drag in agreement with literature.
- Change of momentum computation :
  - Small difference with literature : expected
  - Almost no difference from fitted mesh to adapted mesh $\Rightarrow$ mesh adaptation allow to recover accuracy of fitted mesh.

- MMG platform :
  - Allows to perform every kind of remeshing by given metric field : 2D, 3D, Surface.
  - Surface approximation improvement with MMGS.

Conclusion

- MMG platform :
  - Allows to perform every kind of remeshing by given metric field : 2D, 3D, Surface.
  - Surface approximation improvement with MMGS.
- MMG in progress :
  - Future release of MMG5.1.0 : Volumic and surfacic aniso/iso adaptation.
  - MMG interfaced with Gmsh and FreeFem++.
  - Movement of rigid bodies.

Conclusion

- IBM :
  - Implicit RDS for steady penalized Navier Stokes equations.
  - Explicit 2^{nd} order RK RDS for unsteady Navier Stokes equations.
  - Mesh Adaptation $\Rightarrow$ recovering of the precision of fitted mesh.
  - Mesh Adaptation $\Rightarrow$ improvement of the accuracy of the solution.

Conclusion

- IBM :
  - Implicit RDS for steady penalized Navier Stokes equations.
  - Explicit 2^{nd} order RK RDS for unsteady Navier Stokes equations.
  - Mesh Adaptation $\Rightarrow$ recovering of the precision of fitted mesh.
  - Mesh Adaptation $\Rightarrow$ improvement of the accuracy of the solution.
- IBM in progress :
  - Unsteady mesh adaptation.
  - Level set advection.
  - Objects moved by fluid.

Thank you for your attention